(2)

Technical Report 1394
December 1990

Neural Network Control of an Undersea Manipulator Constrained to Planar Motion

A. W. Westerman

Approved for public release; distribution is unlimited.

NAVAL OCEAN SYSTEMS CENTER
San Diego, California 92152-5000

J. D. FONTANA, CAPT, USN
Commander

H. R. TALKINGTON, Acting
Technical Director

ADMINISTRATIVE INFORMATION

The work reported in this document was done under the Lasers and Microelectronics block program, Independent Exploratory Development (IED) project ZE97, "Neural Network Control of a Robot Arm." The work was done in FY 90 by A. W. Westerman of the Undersea AI and Robotics Branch, NOSC Code 943.

Released by
P. J. Heckman, Head
Undersea AI and Robotics
Branch

Under authority of
N. B. Estabrook, Head
Ocean Engineering Division

LH

SUMMARY

OBJECTIVE

Implement an artificial neural network controller for a manipulator moving in two-dimensional space. The motion was limited to the shoulder and elbow joints. Two tasks were defined within this effort as follows: (1) teach the manipulator to move to a fixed target position from any arbitrary initial position, and (2) teach the manipulator to move to any arbitrary target position from any arbitrary initial position.

RESULTS

Successfully demonstrated the ability to implement a neural network on a manipulator constrained to planar motion. Specifically, the backpropagation network was trained to generalize the manipulator workspace based on a limited number of examples. It correctly generated the appropriate local position control commands to move the arm to an arbitrary target.

RECOMMENDATIONS

As a result of this work, two important issues have become apparent. Follow-on research should address these issues:

1. The first issue became apparent during hardware testing for the arbitrary target case. Gravity has a nonlinear effect on the required torque to move the system and is dependent on the position of the two joints; apparent when the effective angle of the end effector was outside the range of -45 degrees and -135 degrees. This effect would be eliminated if a neural network were trained to learn the dynamics of the arm.

2. The second issue is the effect of a neural network on realtime control. The sampling rate of the system drops dramatically after inclusion of the neural network controller. Test results show the need to implement parallel processing hardware as the complexity of the system and the number of degrees of freedom increases.

CONTENTS

FIGURES

TABLES

INTRODUCTION

The Ocean Engineering Division, Code 94, at the Naval Ocean Systems Center (NOSC) has consistently contributed to the development of undersea vehicles. As the requirement for autonomous underwater vehicles (AUVs) continues to unfold, new and innovative technologies will be necessary to make this long-range requirement a reality. The future of AUVs will be dependent upon the vehicle's ability to perform manipulative tasks, in realtime, in an unstructured underwater environment. Because of these requirements, control algorithms that are adaptable and self-learning are needed.

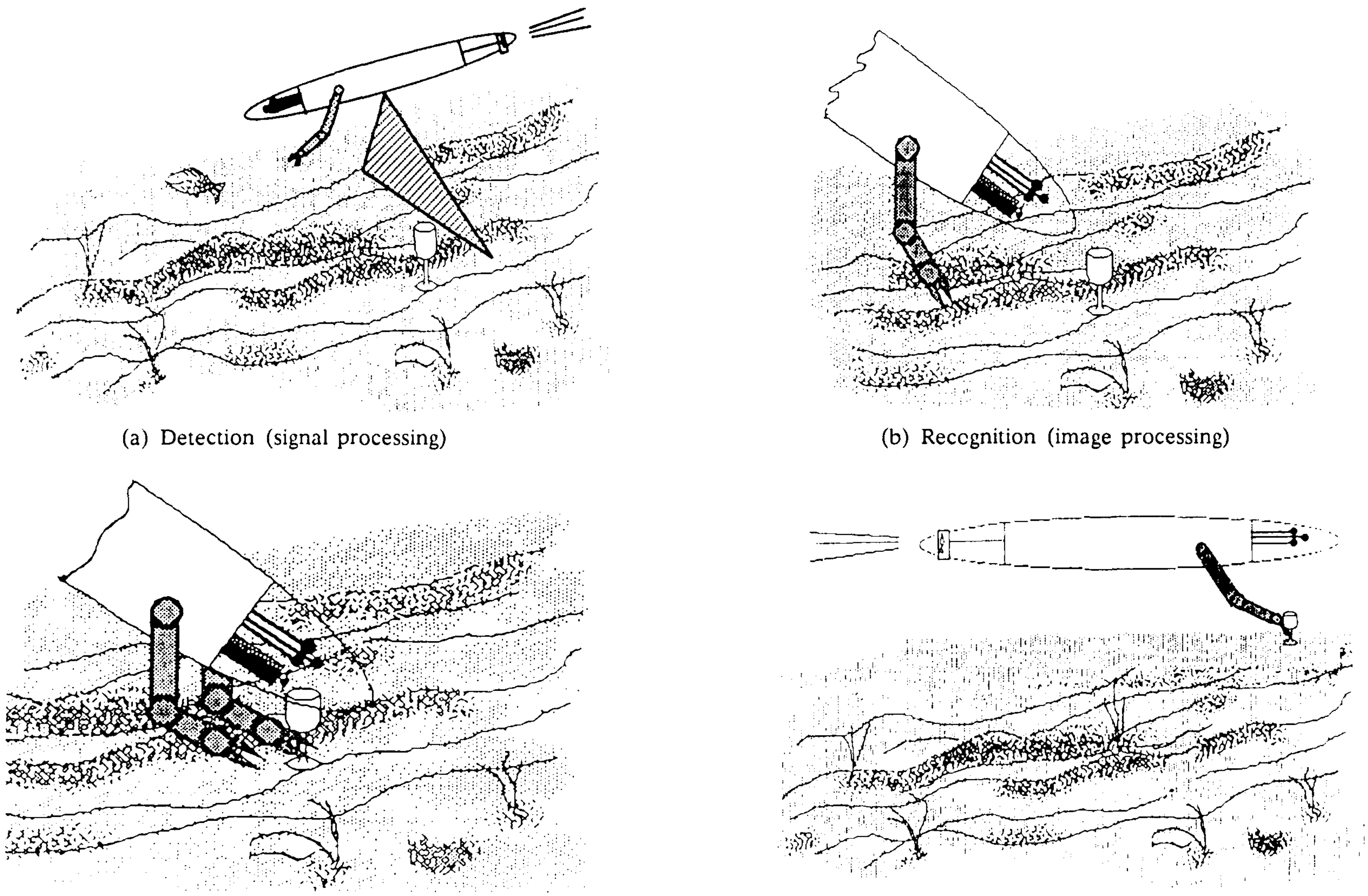
Neural networks are being investigated to address the needs of autonomous work systems required to operate in realtime. For example, figure 1 shows a generic mission scenario for an AUV. As shown, the mission can be divided into three basic subtasks: (1) signal processing for detection; (2) image processing for recognition; and (3) robotics for reaction/interaction. First, the vehicle acts as a search system, scanning the ocean floor for a specified object, in this case a glass. Once the vehicle has detected an object that has the qualities of a glass, the vehicle moves in for a closer look. In this stage, the vehicle uses an imaging system to verify that the object is a glass and further classify it as a particular kind of glass. Finally, based on the object classification, the vehicle picks up the glass and returns with it. Each of these subtasks (signal processing, image processing, and robotics) is being addressed using neural network technology. The merging of all of these disciplines could ultimately provide an autonomous work system. The focus of the work covered in this and future reports will cover the last subtask, that of underwater manipulation.

In 1979, the Undersea AI and Robotics Branch, NOSC Code 943, developed an advanced underwater manipulator to demonstrate new concepts in supervisory control. This arm continues to be an excellent testbed for emerging control technologies. It is an oil-filled, pressure-compensated, dc-direct drive arm with 5 degrees-of-freedom. High torques are achieved using harmonic drive gearing, and position feedback is obtained with potentiometers. The effort discussed in this report is the development and demonstration of a neural network controller on this undersea testbed manipulator.

The long-term objective of this work is to enable the robot arm to perform manipulative tasks in an unstructured ocean environment in realtime with minimal operator intervention. In the ocean, several factors typically make the operational environment difficult to characterize. These may include unanticipated obstacles (e.g., sea life), unexpected external forces (e.g., ocean currents), and poor visibility, which would limit any visual input regarding target location. All of these factors make it virtually impossible to predetermine the operational environment for a particular work task at any given time. This is in contrast to conventional industrial applications, where the workspace tends to be highly structured and predictable.

In addition to a dynamic work environment associated with the ocean, the control of any robot arm presents many challenges. The system dynamics of a typical arm are difficult to model accurately and are highly nonlinear, resulting in many simplifying assumptions to reduce the complexity and number of computations necessary during control. If the dynamics of the manipulator should change, as would occur if system degradation or damage were to take place, the model would no longer be valid. The manipulator would have to be reprogrammed to account for these changes. Reprogramming is undesireable in an underwater application, where the AUV may be submerged for extended periods of time.

Neural networks pose an ideal solution to these problems in that no a priori knowledge about the arm is required. The sensory-motor relationship is learned rather than modelled, thus giving rise to an



(a) Detection (signal processing)

(b) Recognition (image processing)

(c) Reaction/interaction (robotics and vision)

(d) Task completion

Figure 1. Mission scenario for autonomous vehicle.

accurate characterization of the arm. Also, the neural network's inherent adaptability allows it to change and accomodate for any change in, or perturbation to, the system. In addition, the inherently parallel nature of neural networks provides the potential for realtime autonomous control.

BACKGROUND

In conventional robot control, two models are solved. The kinematic model, which is based strictly on the geometry of the arm, determines the mapping between joint space and cartesian space. The inverse of this model, which determines the joint angles required to move to a position in the arm workspace, is of greater interest. The dynamic model, which takes into consideration the physical characteristics of the system (e.g., inertia, friction, damping, and gravity effects of the motors and links), calculates the joint characteristics (acceleration, velocity, position) based on the applied motor torques. The inverse of this model, again being the one of greater interest, maps the desired joint space to a torque space for motor control. This provides the torques required to move the manipulator to a desired point in the workspace based on its current state. Both of these models are required for control, yet their complexity causes them to be computationally expensive. Neural networks have been posed as a solution to eliminate these and other problems with conventional methods.

The application of neural networks to manipulator control is still a field in its infancy. Most of the work in this area has taken place over the last 4 years. A moderate amount of effort has addressed the inverse kinematics problem. A significantly smaller percentage of researchers have been working on the inverse dynamics problem. Some work has also addressed hand-eye coordination through self-learning. Table 1 summarizes the documented work as it relates to manipulator control. Of the work listed in table 1, only five authors have demonstrated their technology on hardware. The remaining work has been devoted to simulations and paper studies. Refer to the bibliography for further information in this area. As can be seen from the table, there are some parallels in the work and associated approaches, which is to be expected in a field this new.

The networks that have been applied to manipulator control include backpropagation, Cerebellar Model Articulation Controller (CMAC), Hopfield, and Kohonen, as well as variations of these approaches. In addition, some work has been done based on biological models, such as Kuperstein's INFANT and Kawato's hierarchical neural network.

Several research areas still need to be addressed. As mentioned in the review article by Horne, Janshidi, and Vadiee (1990), these areas include the following: (1) specifying the network configuration; (2) specifying the teacher; (3) specifying the training set; and (4) specifying the learning algorithm. In other words, the field of neural networks in robotics still needs to answer some fundamental questions. In addition, to achieve realtime capabilities, the neural networks must be readily ported to parallel processing hardware. This is an issue that has not yet been addressed, and yet is a key issue in that the primary advantage of neural networks is their ability to capitalize on the concept of parallel processing. This will be especially true if on-line learning is expected to take place, as is required if a system is to be truly adaptive.

Table 1. Robotics applications in neural networks.

Author	Company/Institution	Network Used	Approach/Application
Barhen, Gulati, Zak	Jet Propulsion Laboratory	Concept of Terminal Attractors	Simulation using constrained configuration of PUMA 560 arm for 3-DOF and 7-DOF redundant manipulators to calculate the inverse kinematic solution.
Bassi, Bekey	University of Southern California (USC) Computer Science Department	Backpropagation	Simulation on 2-DOF manipulator model. Compute inverse dynamics using backpropagation and functional decomposition (decompose dynamic control into three simpler components).
Chen, Pao	Case Western University	Inverse Transfer Learning scheme	Discuss application to 3-DOF arm. No simulation or implementation on hardware. Claim increase convergence over similar approach by Psaltis.
Daunicht	University of Dusseldorf	N/A	Summary paper of neural networks in robotics.
Eckmiller	University of Dusseldorf	Neural Triangular Lattice	Paper study on this neural network model (space-time functions representing trajectories) for internal representation and generation of 2-D movement trajectories. Analysis of "draw what you see" task.
Graf, LaLonde	Carleton University, Canada	Adaptive Arm Controller (AAC) (variation of self-organizing map by Kohonen)	Simulation of results with 2-DOF arm to demonstrate kinematic control for hand-eye coordination. Layers consist of obstacle map, arm configuration map, and eye configuration map.
Guez	Drexel University	Unsupervised Learning (uses some a priori knowledge about arm) (used backpropagation in previous work for inverse kinematics)	Network tested using a simulated 2-DOF planar manipulator. Dynamics represented by weighted linear combination of 3-layer feedforward network modules. Components of the torque equation are learned.
Guo	University of Minnesota	Hopfield	Simulation to solve the inverse kinematic problem for a 4-DOF planar manipulator. Minimize energy function defined as least-squares error (LSE) between actual and desired velocity. (Approach is based on Jacobian control technique.)
Handelman	Princeton	CMAC with rule-based system as teacher and monitor	Simulation of a 2-DOF planar arm. Technique: (1) develop knowledge-based system components to accomplish given control objective; (2) teach neural network by having it observe and generalize; (3) optimize neural network through reinforcement.
Hashimoto	University of Tokyo	Backpropagation	Computer simulation of 6-DOF arm with camera mounted at end-effector. Network generates a change in joint angles such that the coordinates of visual ques. of image coincide with desired ques.
Horne	University of New Mexico	N/A	Summary paper of neural networks in robotics.

Table 1. Robotics applications in neural networks (continued).

Author	Company/Institution	Network Used	Approach/Application
Josin	Neural Systems Inc., Canada	Modified Backpropagation	Simulation of a 2-DOF planar arm. Uses network to develop an inverse kinematic model which calculates the error between desired and actual joint angles.
Kawato, Miyamoto	Osaka University	Hierarchical neural network model	Calculate inverse dynamics using feedback error learning (uses feedback torques to generate the error signal for heterosynaptic learning). Hardware demonstration on a 3-DOF PUMA 260 manipulator.
Lee, Kil	USC	Bidirectional Mapping Neural Network	Simulation of a 6-DOF planar (redundant) manipulator. Network is a backpropagation network with sinusoidal hidden units. A feedback loop is connected to the input to calculate the inverse kinematic solution.
Kuperstein	Neurogen	INFANT	Neural controller designed to learn self-consistency between sensory and motor signals. Inputs are camera images. Outputs are arm signals. Teaches a 5-DOF arm to learn from its own experience guided by visual information from stereo camera.
Martinetz, Ritter, Schulten	University of Munich	Kohonen	Simulated robot arm learns to position its arm and gripper by observing its own movements (inverse kinematic mapping).
Mel	University of Illinois	Sigma-Pi	Neurons act as an interpolating look-up table. Learns by randomly moving through workspace. Control 2-DOF arm planar manipulator with visual input.
Miller	University of New Hampshire	CMAC	Uses CMAC network to map desired object velocities (image space) to actuator drive voltages. Second CMAC network used for image feedback predictions transforming drive voltages to image velocities.
Nguyen	Concordia University	Backpropagation, Backpropagation with error splitting, Functional Link, and Counterpropagation	Simulation of results on 2-DOF planar manipulator. Comparison of networks for learning the forward kinematics of the arm.
Pao, Sobajic	Case Western University	Backpropagation	Network calculates inverse kinematic by mapping current and desired position to change in position. Demonstrate on 2-DOF planar manipulator.
Saxon	Texas A&M	Kohonen	Simulation of a 2-DOF arm which generates a mapping between the workspace (x,y coordinate from camera) and the configuration space (angles of joints to achieve that position).
Werntges	University of Dusseldorf	Backpropagation	Simulation of 2-DOF planar arm. Network trained to learn inverse kinematics using a critic-to-teacher interface.

Table 1. Robotics applications in neural networks (continued).

Author	Company/Institution	Network Used	Approach/Application
Wilhelmsen	University of Utah	Backpropagation, recurrence, error-anticipation	Simulation of controlling a 1-DOF arm rotating in a vertical plane. Network learns inverse dynamics of system.
Yeung	USC	Context sensitive learning network (uses cascaded backpropagation networks)	Simulation of the network learning the inverse Jacobian of PUMA 560 arm.

APPROACH

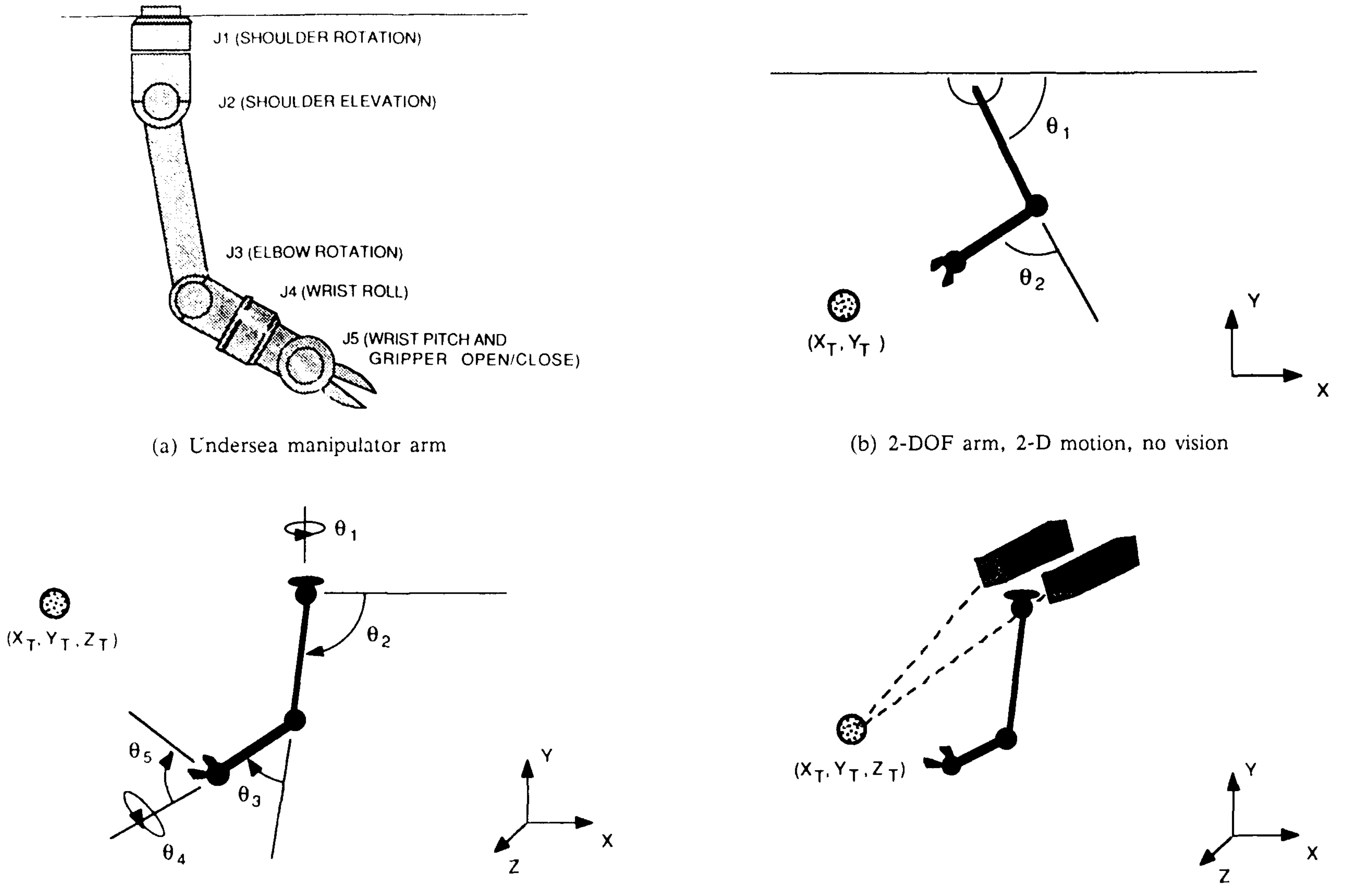
To reach the long-term goal of achieving autonomous manipulative control in an unstructured environment, several simpler steps will be taken. These steps are summarized in figure 2. The focus of the FY 90 effort was to develop a neural network for controlling a 2-degree-of-freedom (2-DOF) manipulator in two-dimensional (2-D) space with no visual input, as shown in figure 2b. This year's tasks were to teach the network to (1) start from any initial position and move towards a fixed target position, and (2) start from an arbitrary initial position and move to an arbitrary target position.

The hardware and software used in this research included the following:

- IBM-compatible 386 machine, 20-MHz clock speed, with 387 math coprocessor chip
- Action Instruments 16-channel multifunction input/output (I/O) card (used for analog-to-digital (A/D) conversion)
- Action Instruments digital-to-analog (D/A) card
- 5-DOF undersea manipulator arm (slave)
- Undersea controller electronics bottle
- Power supply (provides 28 volts dc to motors)
- Master arm

Figure 3 shows the laboratory setup. The 386 workstation with math coprocessor was used for all communication, processing, training, and control. All control software was written in C. The neural network controller for the arm was developed using Olmsted & Watkins (OWL) software, which was chosen for its easy integration with existing software. Figure 4 shows a block diagram of all hardware and associated interconnections. A specification of the manipulator is provided in appendix A.

The approach taken for the FY 90 effort involved training a network to learn an iterative trajectory model and generate the appropriate control signals. A dynamic model is currently being added for learning the required voltage output based on the correction in joint angle. A backpropagation learning algorithm was used for generating the appropriate input-output relationship.



(a) Undersea manipulator arm

(b) 2-DOF arm, 2-D motion, no vision

(c) 5-DOF arm, 3-D motion, no vision

(d) 5-DOF arm, 3-D motion, stereo vision

Figure 2. Autonomous manipulator development.

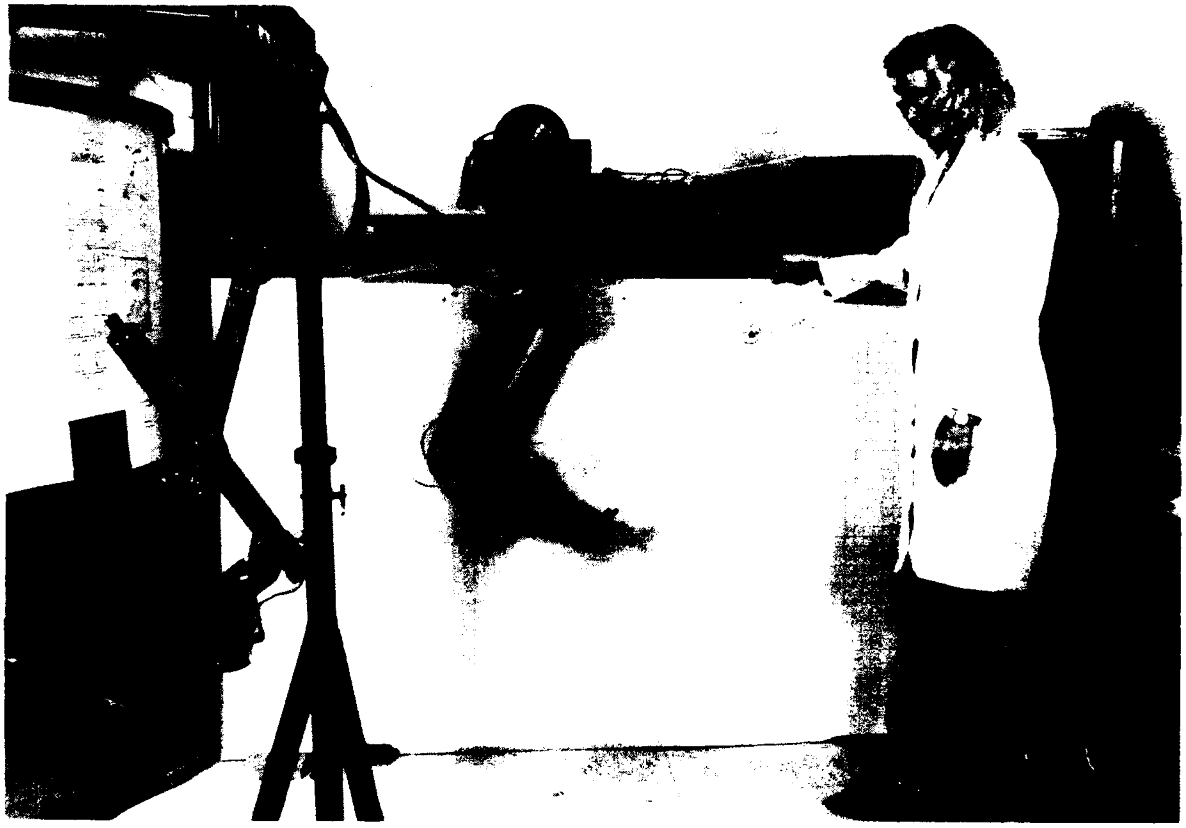


Figure 3. Laboratory setup.

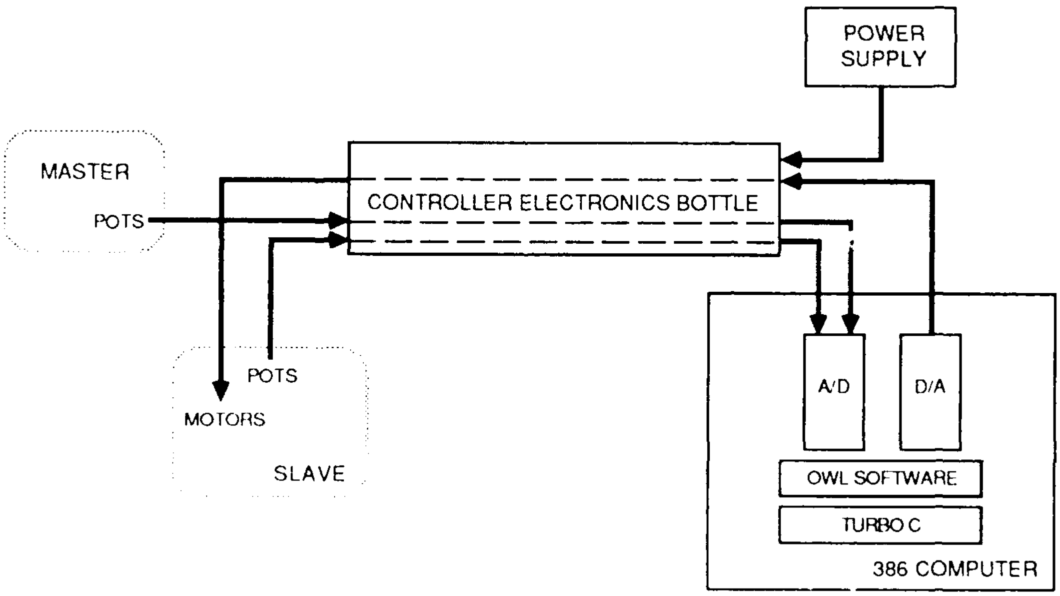


Figure 4. Block diagram of hardware.

FIXED TARGET

The approach taken for the first task (fixed target) was based on the work of Pao and Sobajic (1987). Figure 5 shows the coordinate system. The backpropagation network structure, shown in figure 6, consists of an input layer, two hidden layers, and an output layer. The inputs were the target position designated in polar coordinates (r_t, θ_t) and the current joint angles (θ_1, θ_2). The output from the network was a local action applied to each joint based on these input values and equated to a clockwise or counter-clockwise motion (+1, -1), or no motion (0). The training data for this network is provided in appendix B, along with the resultant weights and thresholds of the trained network. Twenty-nine training patterns were presented during learning. The desired result from the trained network was a generalization of the workspace based on this small set of data points.

The procedure for the fixed target took place in three stages. First, the network was trained off-line using the 29 data sets until a satisfactory sum square error was reached. Second, this network was tested using a simulation program to verify that a smooth trajectory was generated from the initial position to the final position. Third, the network was implemented on the manipulator hardware.

ARBITRARY TARGET

For task 2 (arbitrary target), several changes took place in the training data and the structure of the network. For this application, a set of relative coordinates was chosen to represent the workspace. The outputs from the network remained the same as for the first task, and again represented the local change in joint angle. Figure 7 shows the new coordinate system. Figure 8 shows the resultant back-propagation network consisting of an input layer, a single hidden layer, and an output layer. The inputs to the network were the angle of the elbow joint and the relative angle and distance between the end-effector and the target. Based on these three inputs, a unique combination of control actions for joints 1 and 2 could be represented for the training data. A total of 142 training sets were used for this application. These data are provided in appendix C, along with the resultant weights and thresholds of the trained network. The desired result from the trained network was a generalization of the relative workspace based on a small set of data points.

The procedure for implementation of this network was similar to that for the fixed target. The only difference was that some preprocessing took place before the data were input to the network. This preprocessing provided the relative coordinate information needed by the network. Again, the three stages were off-line training, testing on the simulation software, and implementation on the manipulator.

RESULTS

FIXED TARGET

After a trial-and-error period to determine the optimum learning rate (lr) and momentum (m) terms for convergence, the final parameters were chosen as lr = 0.05 and m = 0.9. The network was then presented the data repeatedly, a single iteration being all 29 sets. After 5000 iterations (15 minutes), the network had learned a complete correlation between input and output space. As per the structure of the OWL software, the neural network was then saved. The network could later be accessed using a library of function calls for both simulation and hardware control.

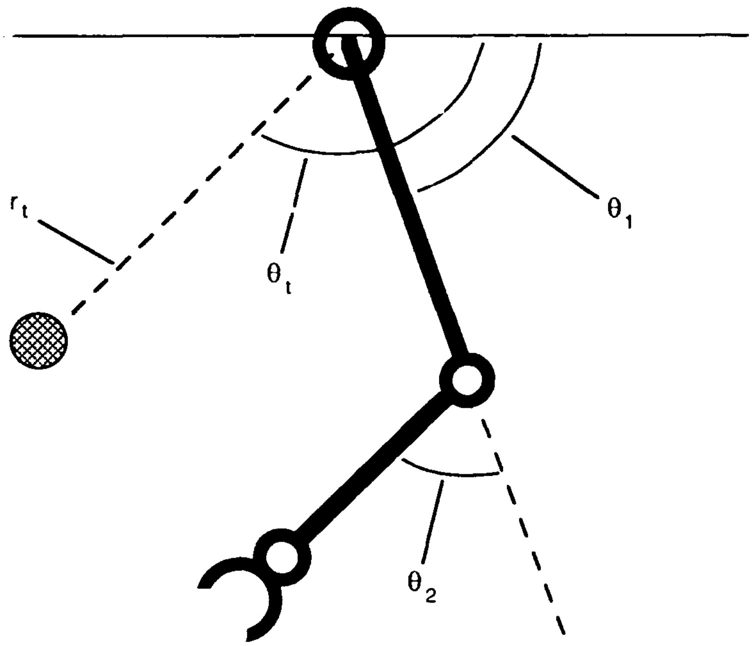


Figure 5. Manipulator model for fixed target.

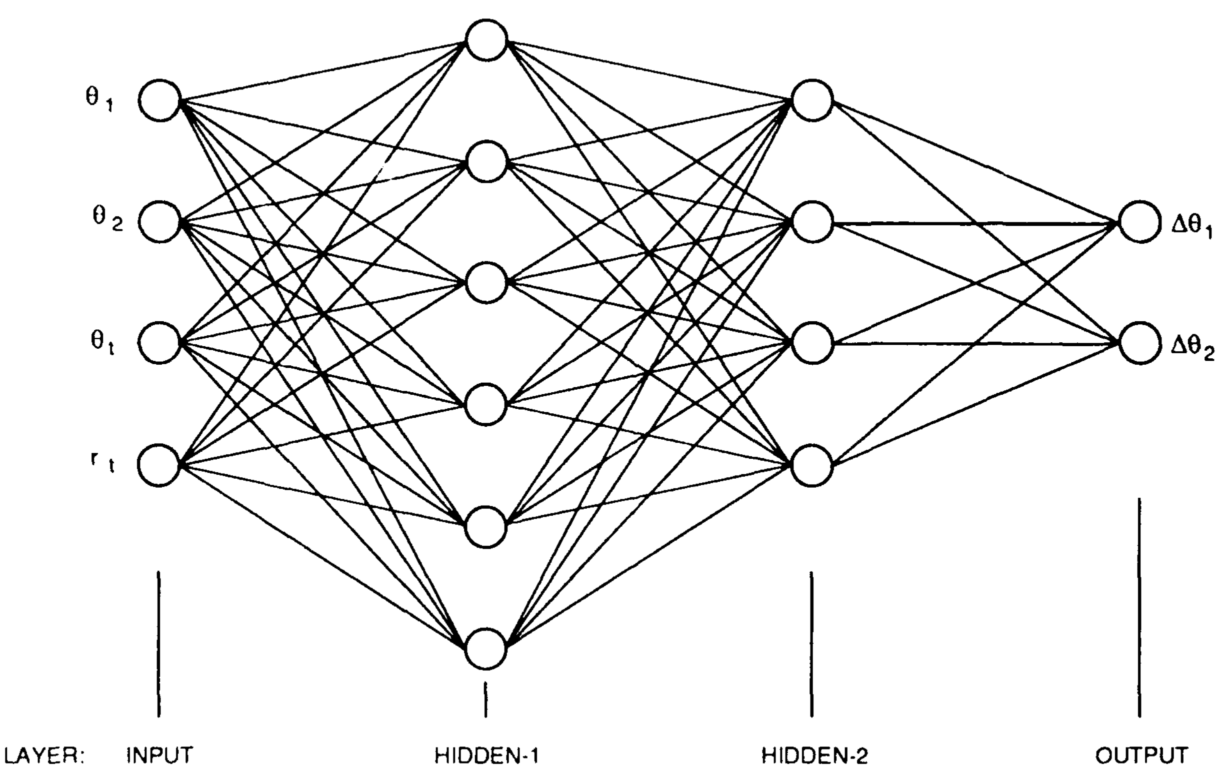


Figure 6. Neural network configuration for fixed target model.

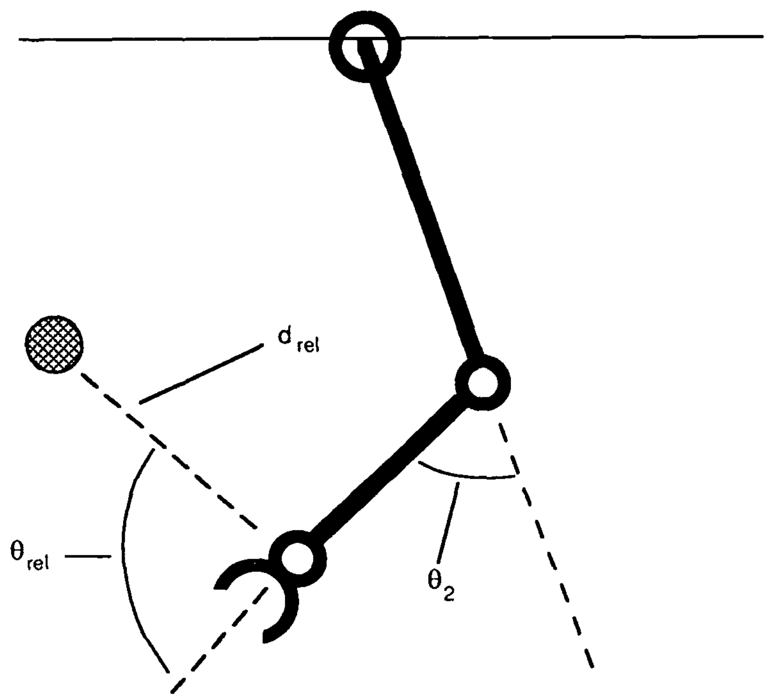


Figure 7. Manipulator model for arbitrary target.

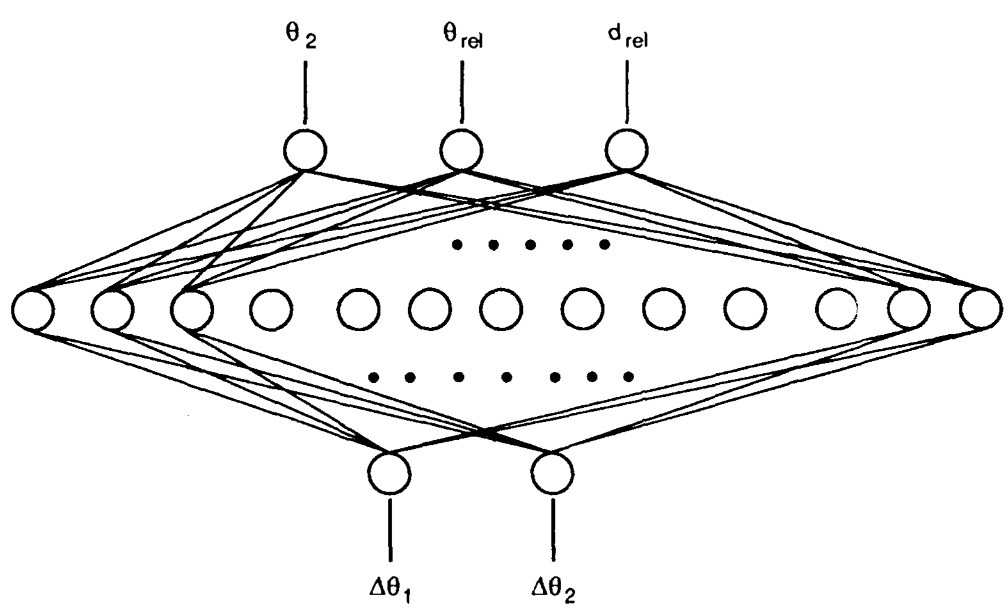


Figure 8. Neural network configuration for arbitrary target model.

The resultant network was tested with the manipulator graphics program, and it was shown that the arm had learned to generalize the workspace based on the initial 29 patterns. The trajectories were always smooth as the arm moved from the initial position to the target. In fact, if a starting position was presented outside the lower half plane, where all training data existed, a smooth trajectory was still generated and the arm moved directly to the target. The final position of the arm was always within 0.5 inch of the target, which equates to 2 percent of the total arm length.

Figure 9 shows the neural network as it is implemented within the manipulator hardware. The current position of the arm is determined by the potentiometer values for the shoulder and elbow joints. The desired target position is specified in polar coordinates and input from the keyboard. These four input values are presented to the trained backpropagation network, and the corresponding local actions are determined for output. The new desired joint positions are updated based on the following relation:

$$\theta_{new_i} = \theta_{old_i} + k * \Delta\theta_i \ .$$

Therefore, each time through the loop, the desired position is updated based on the current position, given by the potentiometer values, and a change in joint angle theta is output from the network. These values are given to the manipulator controller, and the output voltage is generated for each joint.

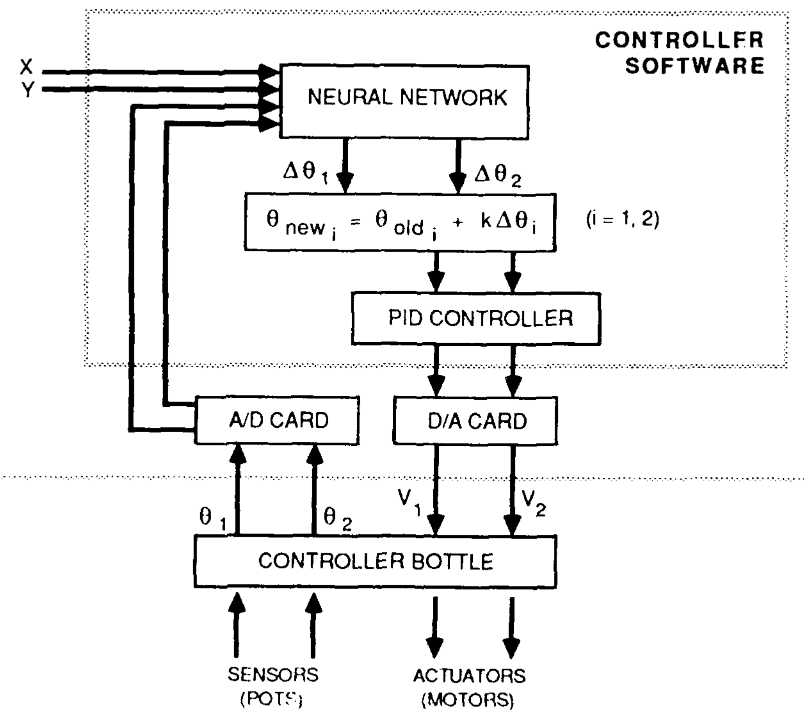


Figure 9. Neural network in relation to hardware and software.

ARBITRARY TARGET

Although the input-output mapping for the fixed target worked well for the stated objective, one limitation made it unsuitable for the arbitrary target. If a target position other than the learned position was input to the network, the arm would still move to the learned target position. The network had generalized the workspace for all initial positions, but the learned joint corrections were based on a single target position.

Building on the results from the fixed target model, a new representation of the data was pursued for the arbitrary target. Rather than provide specific coordinates of the target and arm for training, it was more appropriate to choose relative information about the current state of the system. The relative angle (θ_{rel}) and relative distance (d_{rel}) between the end effector to the target represented the direction the arm should move and the magnitude with which it should move, respectively. By specifying a third input parameter, the elbow position, all ambiguity about the current state of the system is eliminated. As a note, in typical 2-DOF inverse kinematic problems, there are two solutions. By specifying one joint, in this case θ_2, the arm configuration has been uniquely defined. Therefore, the network receives θ_2, θ_{rel}, and d_{rel}, as input.

The output from the network is again a local action for each joint, but in this case both a magnitude and a direction were defined for training. Therefore, large distances would reflect large control actions (+/-1), while small distances would reflect small changes (+/-0.1).

The training of this network was also different than the previous task. Rather than create one large training set which had several d_{rel} values, iterations of smaller data sets were presented for learning. These iterations consisted of varying combinations of θ_2 and θ_{rel}, with d_{rel} fixed. The network was initially trained for $d_{rel} = 24$, then $d_{rel} = 10$, and finally $d_{rel} = 2$. The resultant network commanded large motions at great distances and fine motions at small distances from the target. Each set of data points with fixed d_{rel} was presented 2000 times. The resultant network was tested on the simulation program, and the arm moved smoothly towards the target. However, once the endpoint was within a small tolerance of the target, it would begin to oscillate. This occurred where d_{rel} was very close to zero. The training data was therefore increased from 90 to 142 points, and the resolution of about $d_{rel} = 0$ was improved. This new training set was presented to the previously trained network for 3000 more iterations. The results from this change in the data markedly improved the response of the system and eliminated the oscillation problem around the target.

Once the network was successfully tested in simulation, it was integrated into the hardware, as with the fixed target. Some initial difficulties were apparent with this configuration. The arm would move toward the target and reach the final position if the shoulder angle was between -45 and -135 degrees. However, outside of these limits, where gravity was resisting rather than assisting the control commands, the arm was unable to reach the final position. The arm would reach its static equilibrium position and move no further.

The proportional controller was no longer adequate for this application. The small steps commanded by the network, indicating the arm was close to the target, were not able to overcome the gravitational forces on the system. To solve this problem, a proportional-integral-derivative (PID) controller was developed to replace the proportional controller. The neural network was then retested on the arm and found to move to the target and settle approximately 1.0 inch from the target (4 percent of the arm's total length). Replacing the proportional controller with a PID controller was only a temporary solution to the problem. The more appropriate solution would be to teach the network the dynamics of the system.

CONCLUSIONS AND RECOMMENDATIONS

The work described in this report successfully demonstrated the ability to implement a neural network on a manipulator constrained to planar motion. Specifically, the backpropagation network was able to generalize the manipulator workspace based on a limited number of examples. It correctly generated the appropriate local position control commands to move the arm to an arbitrary target. As a result of this work, however, some important issues have become apparent. It is recommended that the following issues be pursued during follow-on research.

The first issue became apparent during hardware testing for the arbitrary target. Gravity has a nonlinear effect on the required torque to move the system and is dependent on the position of the two joints. This was apparent when the effective angle of the end effector was outside the range of -45 degrees and -135 degrees. This effect could be eliminated if a neural network were trained to learn the dynamics of the arm. Thus, the function of the neural network in the control diagram would change from that shown in figure 10a to the configuration shown in figure 10b.

The goal of this work is to have a system that is adaptable to change. Since most changes would be reflected in the dynamics of the arm, this is a reasonable next step and is currently under investigation.

The second issue is the effect of a neural network on realtime control. Table 2 shows that the sampling rate of the system drops dramatically after inclusion of the neural network controller. The sampling rate drops by a factor of 60. The master/slave controller, which performs mostly integer operations, achieves the sampling limit of the A/D card (3000 Hz). This sampling rate drops to 50 Hz when the neural network is added. If some simple graphics are added to show movement of the arm, this immediately drops to 5 Hz. These results indicate the need to implement parallel processing hardware as the complexity of the system increases (e.g., adding more degrees of freedom, or moving to a three-dimensional space).

Table 2. Variation in sampling rate for undersea manipulator.

Configuration	Sampling Rate
Master/slave only (no graphics or neural network)	3000 Hz
Neural network (no graphics)	50 Hz
Neural network* (with graphics)	5 Hz**

*current arm configuration
**indicates a need to move to parallel processing for FY 91.

Commercial hardware is available for increasing training time, but typically these are dedicated processing boards such as SAIC's* Delta II or Hecht-Neilson's ANZA board. Chips that provide any increase in processing time for a particular application are currently used after training is completed using neural network software. The weights resulting from the trained network are then downloaded onto the neural chip for operational use. Intel's ETANN (Electronically Trainable Artificial Neural Network) chip is being developed to speed up training time, but is still in the developmental stages.

*Science Applications International Corporation (SAIC).

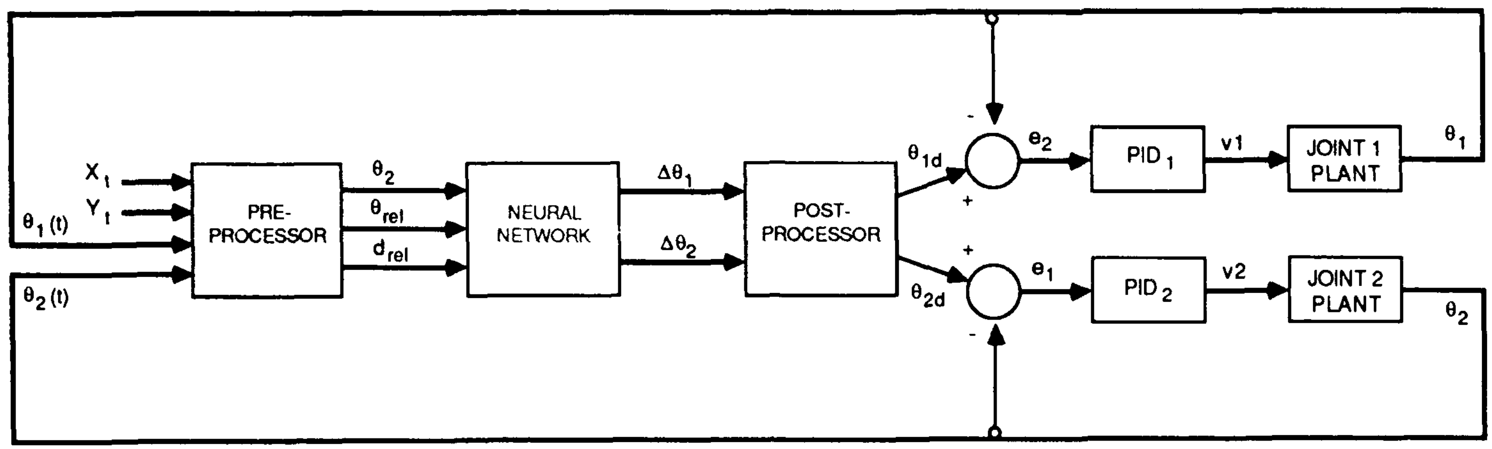


Figure 10a. Present control configuration (iterative trajectory model).

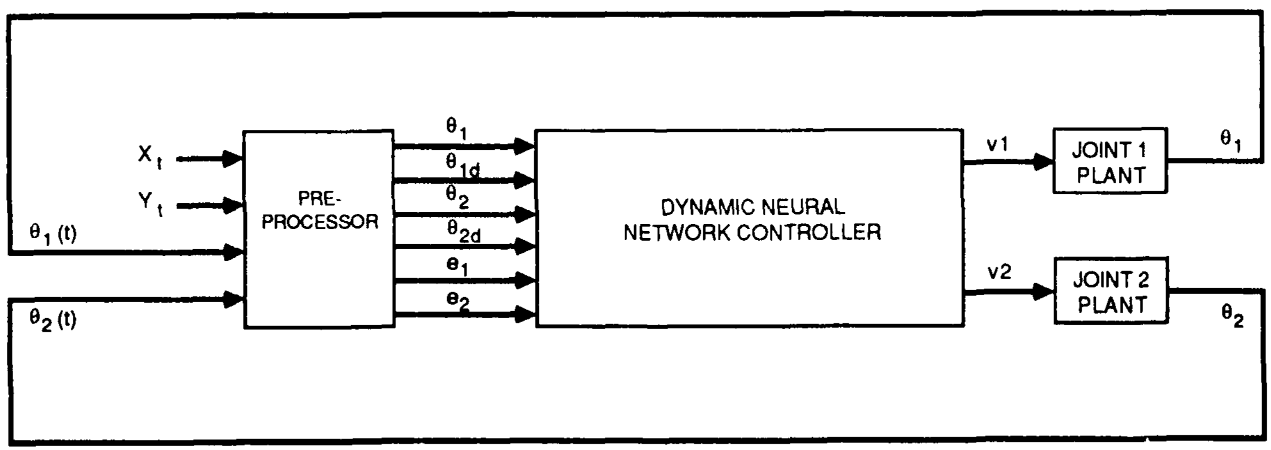


Figure 10b. Proposed dynamic neural network controller.

Processor size becomes an issue when considering its implementation on an autonomous undersea vehicle. In this type of application, large computers cannot be tolerated due to volume constraints of an underwater vehicle. The large computer workstations used in many neural network applications are clearly not a viable solution to the realtime problem for this application. Therefore, implementation of neural network chips will be the method of choice, which will be followed as this technology unfolds.

Both dynamic control and realtime processing will be addressed during FY 91 under the Independent Exploratory Development program at NOSC.

REFERENCES

Horne, B., M. Janshidi, and N. Vadiee. 1990. "Neural Networks in Robotics: A Survey," *J. of Intel. Rob. Systems*, vol. 3, pp. 51-66.

Pao, Y., and D. Sobajic. 1987. "Artificial Neural-Net Based Intelligent Robotics Control," *SPIE Intelligent Robots and Computer Vision*, vol. 848, pp. 542-549.

BIBLIOGRAPHY

Albus, J. S. 1975. "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)," *Trans. of ASME J. of Dyn. Sys. Meas. and Control*, 97, pp. 220-227.

Ali, A. L. 1988. "A Neural Network Approach for Robot Grasp Planning," *Neural Networks*, vol. 1, supp. 1, p. 327.

Anderson, C. W. 1989. "Learning to Control an Inverted Pendulum Using Neural Networks," *IEEE Control Systems Magazine*, pp. 31-37.

Andersson, R. 1989. "Dynamic Sensing in a Ping-Pong Playing Robot," *IEEE Trans. on Robotics and Automation*, vol. 5, no. 6, pp. 728-739.

Andersson, R. 1988. "Investigating Fast, Intelligent Systems with a Ping-Pong Playing Robot," *Robotics Research, 4th Int. Symposium*, pp. 15-22.

Barhen, J., J. R. Einstein, and C. C. Jorgensen. 1987. "Advances in Concurrent Computation for Machine Intelligence and Robotics," *IC7 87: Second Int. Conf. on Supercomputing*, vol. 2, pp. 84-97.

Barhen, J., and S. Gulati. 1989. "Neural Learning of Constrained Nonlinear Transformations," *Computer*, vol. 22, no. 6, pp. 67-76.

Barhen, J., S. Gulati, and M. Zak. 1988. "Real-Time Neuromorphic Algorithms for Inverse Kinematics of Redundant Manipulators," *SPIE Intelligent Robots and Computer Vision*, vol. 1002, pp. 686-696.

Barto, A., C. Anderson, and R. Sutton. 1982. "Synthesis of Nonlinear Control Surfaces by a Layered Associated Search Network," *Biological Cybernetics*, vol. 43, no. 3, pp. 175-185.

Bassi, D. F., and G. Bekey. 1989. "Decomposition of Neural Network Models of Robot Dynamics: A Feasibility Study," *Simulation and AI*, pp. 8-13.

Bassi, D. F., and G. Bekey. 1988. "High Precision Position Control by Cartesian Trajectory Feedback and Connectionist Inverse Dynamics Feedforward," *IEEE Conf. on Neural Networks*, vol. 2, pp. 325-331.

Beard, R. A., and K. Rattan. 1989. "A Neural Network System for Robot Vision," *Proc. 1989 IEEE Nat'l Aerospace and Electronics Conf.*, pp. 1920-1921.

Blackburn, M., and H. Nguyen. 1988. "Biological Model of Vision for an Artificial System that Learns to Perceive Its Environment," *IEEE Conf. on Neural Networks*, vol. 2, pp. 219-225.

Bock, O., and R. Eckmiller. 1986. "Goal-Directed Arm Movements in Absence of Visual Guidance: Evidence for Amplitude Rather Than Position Control," *Exp. Brain Research*, vol. 62, no. 3, pp. 451-458.

Bosse, P., and P. Heckman. 1980. "Development of an Underwater Manipulator for use on a Free-Swimming Unmanned Submersible," NOSC TR 632, Naval Ocean Systems Center, San Diego, CA.

Carpenter, G. 1989. "Neural Network Models for Pattern Recognition and Associative Memory," *Neural Networks*, vol. 2, pp. 243-257.

Chen, V. C., and Y. Pao. 1989. "Learning Control with Neural Networks," *IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 1448-1453.

Cheung, E., and V. Lumelsky. 1989. "Proximity Sensing in Robot Manipulator Motion Planning: System and Implementation Issues," *IEEE Trans. on Robotics and Automation*, vol. 5, no. 6, pp. 740-751.

Dalenoort, G. J. 1984. "On the Representation of Control in Systems in General, and in Neural Networks in Particular," *Cybernetics and Systems Research 2*, pp. 251-255.

Daunicht, W. J. 1989. "Control of Manipulators by Neural Networks," *IEE Proc. Comp. and Digital Techniques*, vol. 136, part E, no. 5, pp. 395-399.

Eckmiller, R. 1988. "Concept of a 4-Joint Machine with Neural Net Control for the Generation of 2-Dimensional Trajectories," *Neural Networks*, vol. 1, supp. 1, p. 334.

Eckmiller, R. 1989. "Neural Nets for Sensory and Motor Trajectories," *IEEE Control Systems Magazine*, pp. 53-59.

Elsley, R. 1988. "A Learning Architecture for Control Based on Backpropagation Neural Networks," *IEEE Conf. on Neural Networks*, vol. 2, pp. 587-594.

Gardner, E. 1988. "What the Robot's Hand Should Tell the Robot's Brain: Feature Detection in a Biological Neural Network," *IEEE Conf. on Neural Networks*, vol. 2, pp. 557-565.

Gawronski, R., and B. Macukow. 1986. "Learning System for Optimal Movement Control with Minimum Error Criterion," RoManSy—Theory and Practice of Robots and Manipulators," *2nd Int'l Symp.*, pp. 253-264. Warsaw, Poland.

Gol'tsev, A. D., and E. Kussul. 1982. "Controlling the Motion of a Moving Robot Using a Neuroid Network," *Trans. Autom. and Remote Control*, vol. 43, no. 5, part 1, pp. 606-611.

Graf, D., and W. LaLonde. 1988. "The Design of an Adaptive Neural Controller for Collision-Free Movement of General Robot Manipulators," *Neural Networks*, vol. 1, supp. 1, p. 335.

Graf, D., and W. LaLonde. 1988. "A Neural Controller for Collision-Free Movement of General Robot Manipulators," *IEEE Conf. on Neural Networks*, pp. 77-84.

Graf, D., and W. Lalonde. 1989. "Neuroplanners for Hand/Eye Coordination," *IJCNN 1989*, vol. 2, pp. 543-548.

Grossberg, S., and J. Marshall. 1989. "Stereo Boundary Fusion by Cortical Complex Cells: A System of Maps, Filters, and Feedback Networks for Multiplexing Distributed Data," *Neural Networks*, vol. 2, pp. 29-51.

Guez, A., and Z. Ahmad. 1988. "Accelerated Convergence in the Inverse Kinematics via Multilayer Feedforward Networks," *IEEE Conf. on Neural Networks*, vol. 2, pp. 341-344.

Guez, A., and Z. Ahmad. 1988. "Solution to the Inverse Kinematics Problem in Robotics by Neural Networks," *Neural Networks*, vol. 1, supp. 1, p. 337.

Guez, A., and J. Selinsky. 1988. "A Neuromorphic Controller with a Human Teacher," *IEEE Conf. on Neural Networks*, vol. 2, pp. 595-602.

Guez, A., and J. Selinsky. 1989. "Neurocontroller Design via Supervised and Unsupervised Learning," *J. of Intelligent and Robotic Systems*, vol. 2, pp. 307-335.

Guo, J., and V. Cherkassky. 1988. "A Solution to the Inverse Kinematic Problem in Robotics Using Neural Network Processing," *IEEE Conf. on Neural Networks*, vol. 2, pp. 299-304.

Handelman, D. A., S. H. Lane, and J. J. Gelfand. 1990. "Integrating Neural Networks and Knowledge-Based Systems for Intelligent Robitic Control," *IEEE Control Systems Magazine*, pp. 77-86.

Hashimoto, H., T. Kubota, M. Sato, and F. Harashima. 1989. "Visual Servo Control of Robotic Manipulators Based on Artificial Neural Network," *IECON*, vol. 1, pp. 770-774.

Helferty, J. J., J. B. Collins, L. C. Wong, and M. Kam. 1989. "A Learning Strategy for the Control of a One-Legged Hopping Robot," *Proc. of 1989 American Control Conf.*, pp. 896-901.

Hewes, R., and W. T. Miller. 1988. "Practical Demonstration of a Learning Control System for a Five-axis Industrial Robot," *SPIE Intelligent Robots and Computer Vision*, vol. 1002, pp. 679-685.

Horne, B., M. Janshidi, and N. Vadiee. 1990. "Neural Networks in Robotics: A Survey," *J. of Intel. Rob. Systems*, vol. 3, pp. 51-66.

Huang, S., and Y. Huang. 1990. "Learning Algorithms for Perceptrons Using Back-Propagation with Selective Updates," *IEEE Control Systems Magazine*, pp. 56-61.

Josin, G., and W. Charney. 1988. "Robot Control Using Neural Networks," *IEEE Conf. on Neural Networks*, vol. 2, pp. 625-631.

Kawato, M., Y. Uno, M. Isobe, and R. Suzuki. 1988. "Hierarchical Neural Network Model for Voluntary Movement with Application to Robotics," *IEEE Control Systems Magazine*, vol. 8, no. 2, pp. 8-15.

Kraft, L. G., and D. P. Campagna. 1990. "A Comparison Between CMAC Neural Network Control and Two Traditional Adaptive Control Systems," *IEEE Control Systems Magazine*, pp. 36-43.

Kung, S., and J. Hwang. 1989. "Neural Network Architectures for Robotic Applications," *IEEE Trans. on Robotics and Automation*, vol. 5, no. 5, pp. 641-657.

Kuperstein, M. 1987. "Adaptive Visual-Motor Coordination in Multi-Joint Robots Using Parallel Architecture," *IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 1595-1601.

Kuperstein, M., and J. Rubinstein. 1988. "Implementation of an Adaptive Neural Controller for Sensory-Motor Coordination, *IEEE Conf. on Neural Networks*, vol. 2, pp. 305-310.

Kuperstein, M., and J. Rubinstein. 1988. "Neural Controller for Adaptive Sensory-Motor Coordination," *SPIE Intelligent Robots and Computer Vision*, vol. 1002, pp. 658-670.

Kuperstein, M. 1988. "Neural Model of Adaptive Hand-Eye Coordination for Single Postures," *Science*, vol. 239, pp. 1308-1311.

Lee, S., and R. M. Kil. 1990. "Robot Kinematic Control Based on Bidirectional Mapping Neural Network," *IJCNN*, vol. 3, pp. 327-335.

Liu, H., T. Iberall, and G. A. Bekey. 1988. "Building a Generic Architecture for Robot Hand Controller," *IEEE Conf. on Neural Networks*, vol. 2, pp. 567-574.

Liu, H., T. Iberall, and G. A. Bekey. 1989. "Neural Network Architecture for Robot Hand Control," *IEEE Control Systems Magazine*, vol. 9, no. 3, pp. 38-42.

Lozano-Perez, T. 1987. "A Simple Motion-Planning Algorithm for General Robot Manipulators," *IEEE J. of Robotics and Automation*, vol. RA-3, no. 3, pp. 224-238.

Martinetz, T. M., H. Ritter, and K. J. Schulten. 1988. "3-D-Neural-Net for Learning Visuomotor Coordination of a Robot Arm," *IEEE Conf. on Neural Networks*, vol. 2, pp. 351-356.

Martinetz, T. M., and K. J. Schulten. 1990. "Hierarchical Neural Net for Learning Control of a Robot's Arm and Gripper," *IJCNN*, vol. II, pp. 747-752.

Matthies, L., and S. Shafer. 1987. "Error Modeling in Stereo Navigation," *IEE J. of Robotics and Automation*, vol. RA-3, no. 3, pp. 239-248.

Mel, B. W. 1988. "Building and Using Mental Models in Sensory-Motor Behavior," *Neural Networks*, vol. 1, supp. 1, p. 346.

Mel, B. W. 1987. "MURPHY: A Robot That Learns By Doing," *IEEE Conf. on Neural Information Processing Systems*.

Miller, W. T. 1987. "Sensor-Based Control of Robotic Manipulators Using a General Learning Algorithm," *IEEE J. of Robotics and Automation*, vol. RA-3, no. 2, pp. 157-165.

Miller W. 1988. "Real-Time Learned Sensor Processing and Motor Control for a Robot with Vision," *Neural Networks*, vol. 1, supp. 1, p. 347.

Miyamoto, H., M. Kawato, T. Setoyama, and R. Suzuki. 1988. "Feedback-Error-Learning Neural Network for Trajectory Control of a Robotic Manipulator," *Neural Networks*, vol. 1, no. 3, pp. 251-265.

Nagata, S., M. Sikiguchi, and K. Asakawa. 1990. "Mobile Robot Control By a Structured Hierarchical Neural Network," *IEEE Control Systems Magazine*, pp. 69-76.

Nguyen, L., R. V. Patel, and K. Khorasani. 1990. "Neural Network Architectures for the Forward Kinematics Problem in Robotics," *IJCNN*, vol. 3, pp. 393-399.

Pabon, J., and D. Gossard. 1988. "The Role of Hidden Layers in Learning Motor Control in Autonomous Systems," *Neural Networks*, vol. 1, supp. 1, p. 351.

Pao, Y., and D. Sobajic. 1987. "Artificial Neural-Net Based Intelligent Robotics Control," *SPIE Intelligent Robots and Computer Vision*, vol. 848, pp. 542-549.

Pourboghrat, F., and M. Sayeh. 1988. "Neural Network Learning Controller for Manipulators," *Neural Networks*, vol. 1, supp. 1, p. 356.

Psaltis, D., A. Sideris, and A. A. Yamamura. 1988. "A Multilayered Neural Network Controller," *IEEE Control Systems Magazine*, pp. 17-20.

Ritter, H. J., T. M. Martinetz, and K. J. Schulten. 1989. "Topology-Conserving Maps for Learning Visuo-Motor-Coordination," *Neural Networks*, vol. 2, pp. 159-168.

Sanner, R., and D. Akin. 1988. "Neuromorphic Regulation of Dynamic Systems Using Back Propagation Networks," *Neural Networks*, vol. 1, supp. 1, p. 358.

Sasseen, R. 1988. "Neurocomputer-Controlled Stepper Motor Balances a Broom," *Power Conversion & Intelligent Motion*, vol. 14, no. 12, pp. 28-32.

Saxon, J. B., and A. Mukerjee. 1990. "Learning the Motion Map of a Robot Arm with Neural Networks," *IJCNN*, vol. 2, pp. 777-782.

Scott, R. W., and D.J. Collins. 1990. "Neural Network Adaptive Controllers," *IJCNN*, vol. 3, pp. 381-386.

Sobajic, D., and P. Lu. 1988. "Intelligent Control of the Intelledex 605T Robot Manipulator," *IEEE Conf. on Neural Networks*, vol. 2, pp. 633-640.

Swiniarski, R. 1990. "A Neuromorphic Cascaded Context Sensitive Controller for Robot Manipulator," *IJCNN*, vol. 3, pp. 387-392.

Tawel, R., and A. Thakoor. 1988. "Neural Networks for Robotic Control," *Neural Networks*, vol. 1, p. 361.

Tolat, V., and B. Widrow. 1988. "An Adaptive Broom Balancer with Visual Inputs," *IEEE Conf. on Neural Networks*, vol. 2, pp. 641-647.

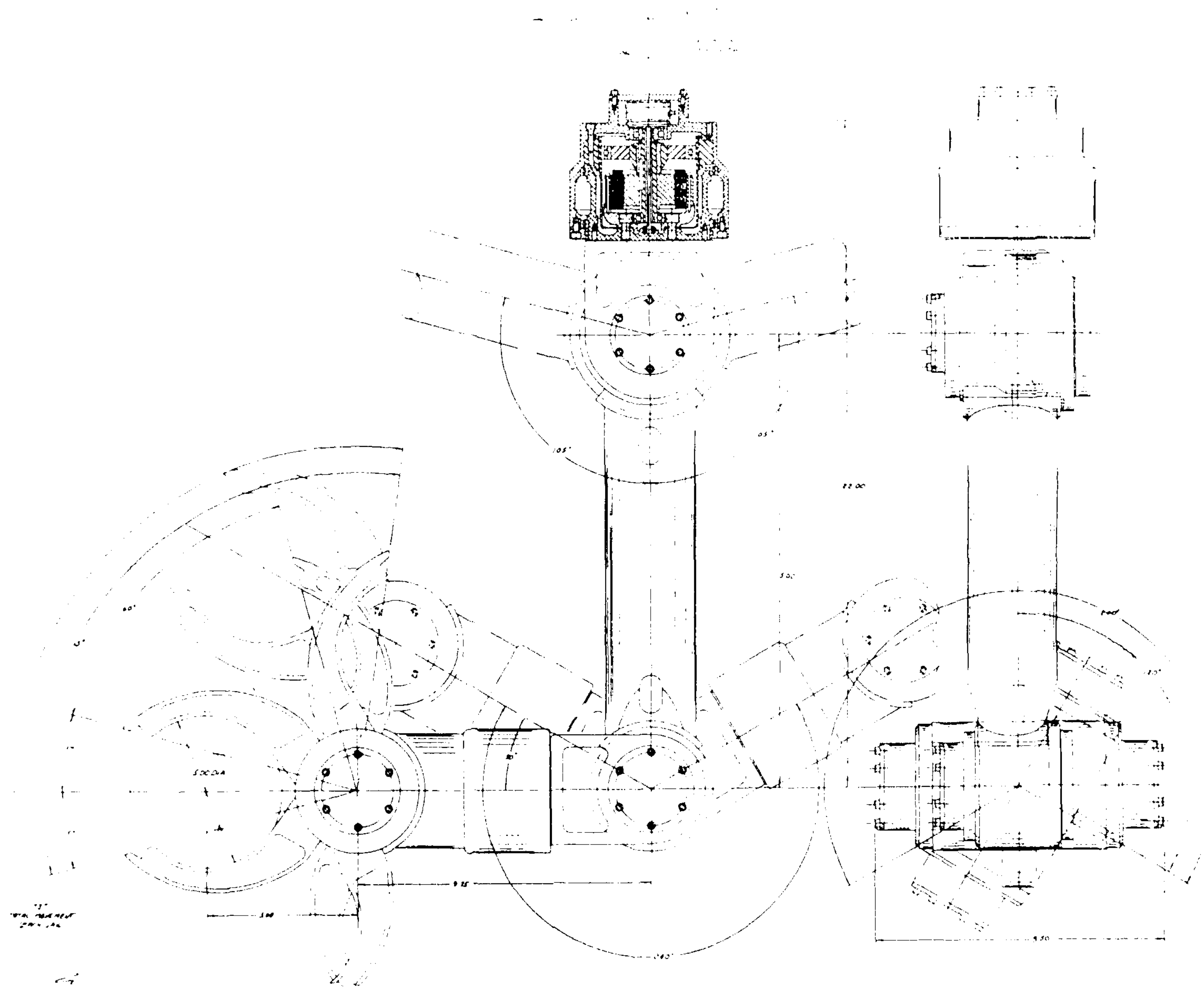
Wang, G. J., and D. K. Miu. 1990. "Unsupervised Adaptive Neural Network Control of Mechanical Systems," *IJCNN*, vol. 3, pp. 421-434.

Werntges, H. W. 1990. "Delta Rule-Based Neural Networks for Inverse Kinematics: Error Gradient Reconstruction Replaces the Teacher," *IJCNN*, vol. 3, p. 415.

Wilhelmsen, K., and N. Cotter. 1990. "Neural Network Based Controllers for a Single-Degree-of-Freedom Robotic Arm," *IJCNN*, pp. 407-413.

APPENDIX A

MANIPULATOR ASSEMBLY



Manipulator assembly drawing showing the movement angle for each joint (Bosse, Heckman, 1973).

APPENDIX B

TRAINING DATA FOR FIXED TARGET

```
   Channel 1 REPORT

Network:  bkp
Date:     Wed Jul 25 16:48:16 1990

Instantiation Parameters:
    Input width: 4
    Layers NOT counting input: 3
    Connect inputs to outputs: NO
    Enable batching and/or momentum: YES

Layer 1:    6 states    30 weights

    States:
4.80632e-05    0.386575    0.999955 1.45762e-06    0.368962    0.0105554

    Weights:
 -0.269475    2.05898   0.0997021    9.37384    -10.0902    -5.92577
 -0.336167    3.44077    -4.37543    12.8207     2.39504    -2.49367
 -0.721259    1.95454     25.6299    -3.3044     1.51174    0.926647
   12.4374    -4.1165     7.35644   0.592935    -2.96981     17.8781
   7.63579    4.98866    0.502324   -2.07669     21.8219     11.9664
Layer 2:    4 states    28 weights

    States:
2.87064e-05 2.03691e-11  0.00185027   0.443627

    Weights:
  -10.0009    2.17331    -6.59265    8.76634    -2.07684     -17.906
  -6.45688   -14.2998     8.57879    -2.3836    -10.9409   -0.551944
    3.9099     9.6293     8.36337   -16.6467       16.37    -24.1206
  -7.67519    8.67966    -6.10177    6.07905    -5.63288     4.14077
  -5.91955   -2.27655    -5.33978   -1.07944
Layer 3:    2 states    10 weights

    States:
 -0.0980462    0.992683

    Weights:
  -1.04181    6.50194     1.03915  -0.134066     2.13326    0.997596
 -0.318167  -0.997277    -2.08652  0.0193913
```

28

20.000	-90.000	0.000	-30.000	1.000	-1.000
20.000	-90.000	0.000	-60.000	0.000	-1.000
20.000	-90.000	0.000	-90.000	-1.000	0.000
20.000	-90.000	-30.000	30.000	-1.000	1.000
20.000	-90.000	-30.000	-30.000	0.000	-1.000
20.000	-90.000	-30.000	-60.000	0.000	-1.000
20.000	-90.000	-30.000	-90.000	-1.000	0.000
20.000	-90.000	-60.000	60.000	-1.000	1.000
20.000	-90.000	-60.000	-30.000	0.000	-1.000
20.000	-90.000	-60.000	-60.000	0.000	-1.000
20.000	-90.000	-60.000	-90.000	0.000	1.000
20.000	-90.000	-90.000	90.000	-1.000	0.000
20.000	-90.000	-90.000	30.000	-1.000	1.000
20.000	-90.000	-90.000	-30.000	1.000	-1.000
20.000	-90.000	-90.000	-90.000	1.000	0.000
20.000	-90.000	-120.000	90.000	0.000	-1.000
20.000	-90.000	-120.000	30.000	0.000	1.000
20.000	-90.000	-120.000	-60.000	1.000	-1.000
20.000	-90.000	-150.000	90.000	1.000	0.000
20.000	-90.000	-60.000	-77.500	0.000	0.000
20.000	-90.000	-120.000	77.500	0.000	0.000
20.000	-90.000	-180.000	30.000	-1.000	1.000
20.000	-90.000	-180.000	60.000	0.000	1.000
20.000	-90.000	-180.000	90.000	1.000	0.000
20.000	-90.000	-150.000	-30.000	1.000	-1.000
20.000	-90.000	-150.000	30.000	0.000	1.000
20.000	-90.000	-150.000	60.000	0.000	1.000
20.000	-90.000	-120.000	60.000	0.000	1.000

APPENDIX C

TRAINING DATA FOR ARBITRARY TARGET

Channel 1 REPORT

Network: bkp
Date: Wed Jul 25 16:51:52 1990

Instantiation Parameters:
Input width: 3
Layers NOT counting input: 2
Connect inputs to outputs: NO
Enable batching and/or momentum: YES

Layer 1: 15 states 60 weights

States:

0.128261	0.957379	0.0654994	0.00619002	0.0569937	0.108257
0.116211	0.169312	0.0102753	0.129485	0.107508	0.420413
0.268152	0.842259	0.121647			

Weights:

-1.91642	-0.434545	-0.219915	-1.03742	3.11185	-3.52452
-6.71229	-3.51278	-2.65797	-1.98118	6.32806	-6.94234
-5.07861	4.79581	6.34193	-3.3764	-2.80613	0.461364
6.68677	-3.74042	-2.10867	-1.60066	0.933791	1.17689
-2.02881	-0.938167	0.0143177	-0.122049	-1.59051	-0.337332
-0.623784	-0.790624	-4.56769	2.23613	5.92454	-6.38177
-1.90552	-1.20614	0.00795496	-0.292831	-2.11645	-0.0310354
-1.13043	-1.12275	-0.321079	-2.45533	-2.48326	-5.09297
-1.00402	-5.5448	7.44622	-1.68711	1.67513	1.37986
-8.16609	-2.79071	-1.97693	-0.809286	-0.183751	-0.257245

Layer 2: 2 states 32 weights

States:
-2.85941e-07 5.42348e-07

Weights:

0.0368198	-0.433293	3.55933	3.40084	3.10685	-0.127453
1.58095	0.333807	-0.205273	-3.9535	0.529029	-0.842791
-4.16339	-2.24248	-1.66346	0.137121	-1.08684	-0.14084
-0.147521	-2.86126	-1.99677	4.44787	-0.492823	-0.279278
0.151702	0.601137	-0.284276	0.56043	-2.03128	1.08882
2.14516	-0.179769				

```
142 3 2
-115.00 180.00  2.00 1.00 0.00
-115.00 150.00  2.00 1.00 1.00
-115.00 120.00  2.00 1.00 1.00
-115.00 90.00  2.00 0.00 1.00
-115.00 60.00  2.00 -1.00 1.00
-115.00 0.00  2.00 -1.00 1.00
-115.00 -23.00  2.00 -1.00 0.00
-115.00 -45.00  2.00 -1.00 -1.00
-115.00 -60.00  2.00 -1.00 -1.00
-115.00 -90.00  2.00 0.00 -1.00
-115.00 -120.00  2.00 1.00 -1.00
-115.00 -150.00  2.00 1.00 0.00
-115.00 -180.00  2.00 1.00 0.00
-115.00 150.00 0.00 0.00 0.00
-115.00 120.00 0.00 0.00 0.00
-115.00 90.00 0.00 0.00 0.00
-115.00 60.00 0.00 0.00 0.00
-115.00 30.00 0.00 0.00 0.00
-115.00 0.00 0.00 0.00 0.00
-115.00 -150.00 0.00 0.00 0.00
-115.00 -120.00 0.00 0.00 0.00
-115.00 -90.00 0.00 0.00 0.00
-115.00 -60.00 0.00 0.00 0.00
-115.00 -30.00 0.00 0.00 0.00
-90.00 180.00  2.00 1.00 -1.00
-90.00 150.00  2.00 1.00 0.00
-90.00 120.00  2.00 1.00 1.00
-90.00 90.00  2.00 0.00 1.00
-90.00 60.00  2.00 -1.00 1.00
-90.00 -30.00  2.00 -1.00 1.00
-90.00 -45.00  2.00 -1.00 0.00
-90.00 -60.00  2.00 -1.00 -1.00
-90.00 -90.00  2.00 0.00 -1.00
-90.00 -120.00  2.00 1.00 -1.00
-90.00 -180.00  2.00 1.00 -1.00
-90.00 150.00 0.00 0.00 0.00
-90.00 120.00 0.00 0.00 0.00
-90.00 90.00 0.00 0.00 0.00
-90.00 60.00 0.00 0.00 0.00
-90.00 30.00 0.00 0.00 0.00
-90.00 0.00 0.00 0.00 0.00
-90.00 -150.00 0.00 0.00 0.00
-90.00 -120.00 0.00 0.00 0.00
-90.00 -90.00 0.00 0.00 0.00
-90.00 -60.00 0.00 0.00 0.00
-90.00 -30.00 0.00 0.00 0.00
-45.00 150.00  2.00 1.00 -1.00
-45.00 135.00  2.00 1.00 0.00
-45.00 120.00  2.00 1.00 1.00
-45.00 90.00  2.00 0.00 1.00
-45.00 60.00  2.00 -1.00 1.00
-45.00 -45.00  2.00 -1.00 1.00
-45.00 -60.00  2.00 -1.00 0.00
-45.00 -75.00  2.00 -0.75 -1.00
-45.00 -90.00  2.00 0.00 -1.00
-45.00 -120.00  2.00 1.00 -1.00
-45.00 150.00 0.00 0.00 0.00
-45.00 120.00 0.00 0.00 0.00
-45.00 90.00 0.00 0.00 0.00
-45.00 60.00 0.00 0.00 0.00
-45.00 30.00 0.00 0.00 0.00
-45.00 0.00 0.00 0.00 0.00
-45.00 -150.00 0.00 0.00 0.00
-45.00 -120.00 0.00 0.00 0.00
-45.00 -90.00 0.00 0.00 0.00
-45.00 -60.00 0.00 0.00 0.00
-45.00 -30.00 0.00 0.00 0.00
0.00 90.00  2.00 0.00 1.00
0.00 -90.00  2.00 0.00 -1.00
0.00 150.00 0.00 0.00 0.00
0.00 120.00 0.00 0.00 0.00
0.00 90.00 0.00 0.00 0.00
0.00 -150.00 0.00 0.00 0.00
0.00 -120.00 0.00 0.00 0.00
0.00 -90.00 0.00 0.00 0.00
```

```
0.00 0.00 0.00 0.00 0.00
45.00 120.00  2.00 -1.00 1.00
45.00 90.00  2.00 0.00 1.00
45.00 75.00  2.00 0.50 1.00
45.00 60.00  2.00 1.00 0.00
45.00 45.00  2.00 1.00 -1.00
45.00 -60.00  2.00 1.00 -1.00
45.00 -90.00  2.00 0.00 -1.00
45.00 -120.00  2.00 -1.00 -1.00
45.00 -135.00  2.00 -1.00 0.00
45.00 -150.00  2.00 -1.00 1.00
45.00 150.00 0.00 0.00 0.00
45.00 120.00 0.00 0.00 0.00
45.00 90.00 0.00 0.00 0.00
45.00 60.00 0.00 0.00 0.00
45.00 30.00 0.00 0.00 0.00
45.00 0.00 0.00 0.00 0.00
45.00 -150.00 0.00 0.00 0.00
45.00 -120.00 0.00 0.00 0.00
45.00 -90.00 0.00 0.00 0.00
45.00 -60.00 0.00 0.00 0.00
45.00 -30.00 0.00 0.00 0.00
90.00 180.00  2.00 -1.00 1.00
90.00 120.00  2.00 -1.00 1.00
90.00 90.00  2.00 0.00 1.00
90.00 60.00  2.00 1.00 1.00
90.00 45.00  2.00 1.00 0.00
90.00 30.00  2.00 1.00 -1.00
90.00 -60.00  2.00 1.00 -1.00
90.00 -90.00  2.00 0.00 -1.00
90.00 -120.00  2.00 -1.00 -1.00
90.00 -150.00  2.00 -1.00 0.00
90.00 -180.00  2.00 -1.00 1.00
90.00 150.00 0.00 0.00 0.00
90.00 120.00 0.00 0.00 0.00
90.00 90.00 0.00 0.00 0.00
90.00 60.00 0.00 0.00 0.00
90.00 30.00 0.00 0.00 0.00
90.00 0.00 0.00 0.00 0.00
90.00 -150.00 0.00 0.00 0.00
90.00 -120.00 0.00 0.00 0.00
90.00 -90.00 0.00 0.00 0.00
90.00 -60.00 0.00 0.00 0.00
90.00 -30.00 0.00 0.00 0.00
115.00 180.00  2.00 -1.00 0.00
115.00 150.00  2.00 -1.00 1.00
115.00 120.00  2.00 -1.00 1.00
115.00 90.00  2.00 0.00 1.00
115.00 60.00  2.00 1.00 1.00
115.00 45.00  2.00 1.00 1.00
115.00 23.00  2.00 1.00 0.00
115.00 0.00  2.00 1.00 -1.00
115.00 -60.00  2.00 1.00 -1.00
115.00 -90.00  2.00 0.00 -1.00
115.00 -150.00  2.00 -1.00 -1.00
115.00 -180.00  2.00 -1.00 0.00
115.00 150.00 0.00 0.00 0.00
115.00 120.00 0.00 0.00 0.00
115.00 90.00 0.00 0.00 0.00
115.00 60.00 0.00 0.00 0.00
115.00 30.00 0.00 0.00 0.00
115.00 0.00 0.00 0.00 0.00
115.00 -150.00 0.00 0.00 0.00
115.00 -120.00 0.00 0.00 0.00
115.00 -90.00 0.00 0.00 0.00
115.00 -60.00 0.00 0.00 0.00
115.00 -30.00 0.00 0.00 0.00
```

```
90 3 2
-115.00 180.00 2.00 1.00 0.00
-115.00 150.00 2.00 1.00 1.00
-115.00 120.00 2.00 1.00 1.00
-115.00 90.00 2.00 0.00 1.00
-115.00 60.00 2.00 -1.00 1.00
-115.00 0.00 2.00 -1.00 1.00
-115.00 -23.00 2.00 -1.00 0.00
-115.00 -45.00 2.00 -1.00 -1.00
-115.00 -60.00 2.00 -1.00 -1.00
-115.00 -90.00 2.00 0.00 -1.00
-115.00 -120.00 2.00 1.00 -1.00
-115.00 -150.00 2.00 1.00 0.00
-115.00 -180.00 2.00 1.00 0.00
-115.00 90.00 0.00 0.00 0.00
-115.00 0.00 0.00 0.00 0.00
-115.00 -90.00 0.00 0.00 0.00
-90.00 180.00 2.00 1.00 -1.00
-90.00 150.00 2.00 1.00 0.00
-90.00 120.00 2.00 1.00 1.00
-90.00 90.00 2.00 0.00 1.00
-90.00 60.00 2.00 -1.00 1.00
-90.00 -30.00 2.00 -1.00 1.00
-90.00 -45.00 2.00 -1.00 0.00
-90.00 -60.00 2.00 -1.00 -1.00
-90.00 -90.00 2.00 0.00 -1.00
-90.00 -120.00 2.00 1.00 -1.00
-90.00 -180.00 2.00 1.00 -1.00
-90.00 90.00 0.00 0.00 0.00
-90.00 0.00 0.00 0.00 0.00
-90.00 -90.00 0.00 0.00 0.00
-45.00 150.00 2.00 1.00 -1.00
-45.00 135.00 2.00 1.00 0.00
-45.00 120.00 2.00 1.00 1.00
-45.00 90.00 2.00 0.00 1.00
-45.00 60.00 2.00 -1.00 1.00
-45.00 -45.00 2.00 -1.00 1.00
-45.00 -60.00 2.00 -1.00 0.00
-45.00 -75.00 2.00 -0.75 -1.00
-45.00 -90.00 2.00 0.00 -1.00
-45.00 -120.00 2.00 1.00 -1.00
-45.00 90.00 0.00 0.00 0.00
-45.00 0.00 0.00 0.00 0.00
-45.00 -90.00 0.00 0.00 0.00
0.00 90.00 2.00 0.00 1.00
0.00 -90.00 2.00 0.00 -1.00
0.00 90.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00
0.00 -90.00 0.00 0.00 0.00
45.00 120.00 2.00 -1.00 1.00
45.00 90.00 2.00 0.00 1.00
45.00 75.00 2.00 0.50 1.00
45.00 60.00 2.00 1.00 0.00
45.00 45.00 2.00 1.00 -1.00
45.00 -60.00 2.00 1.00 -1.00
45.00 -90.00 2.00 0.00 -1.00
45.00 -120.00 2.00 -1.00 -1.00
45.00 -135.00 2.00 -1.00 0.00
45.00 -150.00 2.00 -1.00 1.00
45.00 90.00 0.00 0.00 0.00
45.00 0.00 0.00 0.00 0.00
45.00 -90.00 0.00 0.00 0.00
90.00 180.00 2.00 -1.00 1.00
90.00 120.00 2.00 -1.00 1.00
90.00 90.00 2.00 0.00 1.00
90.00 60.00 2.00 1.00 1.00
90.00 45.00 2.00 1.00 0.00
90.00 30.00 2.00 1.00 -1.00
90.00 -60.00 2.00 1.00 -1.00
90.00 -90.00 2.00 0.00 -1.00
90.00 -120.00 2.00 -1.00 -1.00
90.00 -150.00 2.00 -1.00 0.00
90.00 -180.00 2.00 -1.00 1.00
90.00 90.00 0.00 0.00 0.00
90.00 0.00 0.00 0.00 0.00
90.00 -90.00 0.00 0.00 0.00
```

```
115.00 180.00 2.00 -1.00 0.00
115.00 150.00 2.00 -1.00 1.00
115.00 120.00 2.00 -1.00 1.00
115.00 90.00 2.00 0.00 1.00
115.00 60.00 2.00 1.00 1.00
115.00 45.00 2.00 1.00 1.00
115.00 23.00 2.00 1.00 0.00
115.00 0.00 2.00 1.00 -1.00
115.00 -60.00 2.00 1.00 -1.00
115.00 -90.00 2.00 0.00 -1.00
115.00 -150.00 2.00 -1.00 -1.00
115.00 -180.00 2.00 -1.00 0.00
115.00 90.00 0.00 0.00 0.00
115.00 0.00 0.00 0.00 0.00
115.00 -90.00 0.00 0.00 0.00
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
December 1990

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
NEURAL NETWORK CONTROL OF AN UNDERSEA MANIPULATOR CONSTRAINED TO PLANAR MOTION

5. FUNDING NUMBERS
PE: 0602936N
WU: DN300022

6. AUTHOR(S)
A. W. Westerman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Ocean Systems Center
San Diego, CA 92152-5000

8. PERFORMING ORGANIZATION REPORT NUMBER
NOSC TR 1394

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Chief of Naval Research
Independent Exploratory Development Programs
Arlington, VA 22217

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report discusses the development of an artificial neural network controller for a manipulator moving in two-dimensional space. The motion was limited to the shoulder and elbow joints. Two tasks were defined within this effort as follows: (1) teach the manipulator to move to a fixed target position from any arbitrary initial position, and (2) teach the manipulator to move to any arbitrary target position from any arbitrary initial position. The backpropagation learning/algorithm was implemented for both subtasks. The trained neural network was initially tested using a graphics simulation program and late transferred to the manipulator hardware.

14. SUBJECT TERMS
artificial neural network controller

15. NUMBER OF PAGES
40

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAME AS REPORT

INITIAL DISTRIBUTION

Code 0012	Patent Counsel	(1)
Code 014	Dr. W. Rasmussen	(1)
Code 0142	K. Campbell	(1)
Code 90	I. Lemaire	(1)
Code 94	N. Estabrook	(1)
Code 943	P. Heckman	(1)
Code 943	A. Westerman	(30)
Code 952B	J. Puleo	(1)
Code 961	Archive/Stock	(6)
Code 964B	Library	(3)

Defense Technical Information Center
Alexandria, VA 22302-6145 (4)

NOSC Liaison Office
Washington, DC 20363-5100

Center for Naval Analyses
Alexandria, VA 22302-0268